REMARKS

The above amendments and following remarks attend to every rejection and issue presented in the pending July 11, 2005 office action.

Claim 1 is amended to recite that text strings associated with chassis logs for one entity of the electronic architecture are processed and that sequences of text strings are transformed into human interpretable statements. Support for this amendment is found in at least paragraph [0028] of the specification.

Claim 5 is amended to clarify the step of processing text strings. As disclosed in paragraph [0028], chassis codes are converted to strings, and the analyzers analyze these text strings associated with an entity, thereby processing the chassis codes.

Claim 15 is amended to correct a typographical error and to recite analyzing sequences of text strings. Support for this amendment is found at least in paragraph [0028].

No new matter is added. Claims 1-20 remain pending, with claims 1 and 15 being independent.

Claim Rejections – 35 U.S.C. §103(a)

Claims 1-4 and 6-20 stand rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,524,245 granted to Rock et al. (hereinafter "Rock") in view of U.S. Patent No. 5,724,503 granted to Kleinman et al. (hereinafter "Kleinman"). Respectfully, we disagree.

By way of background, the immediate application teaches a method and system for analyzing events from electronic architecture and for generating human interpretable statements summarizing at least one of these events. These events take the form of chassis codes that are produced by one or more entities of the electronic architecture, to indicate status of the entity during boot and operation. These chassis codes do not necessarily indicate a failure or problem; they are not therefore 'error codes'. Chassis codes are extracted from an electronic architecture, filtered and converted to text strings. These text strings are sent to an analyzer associated with the entity that produced the chassis codes. See paragraph [0028] of the specification and FIG. 2. Each analyzer is associated with an entity of the electronic architecture, and

processes text strings associated with chassis codes for its entity. In processing the text strings, each analyzer recognizes one or more sequences of chassis codes to identify a particular problem of the associated entity. These text strings are transferred into one or more human interpretable statements summarizing at least one problem detail of the entity. See at least paragraph [0034] of the specification and FIG. 4A and FIG. 4B.

On the other hand, Rock discloses responding to error messages from medical diagnostic ultrasound imaging systems. Rock does not process chassis codes that indicate entity health for electronic architecture. Rock receives only error messages from an ultrasound imaging system that occur when a fault is detected within the ultrasound imaging system. Further, Rock does not process sequences of error messages to recognize error code sequences that identify a particular problem; Rock processes one error code at a time. The error messages of Rock are generated and sent from the ultrasound imaging system to a processor for processing. Rock therefore does not 'extract' chassis codes from an electronic architecture as taught by the immediate application. The error message of Rock can contain one or more of: "the application in the ultrasound system in which the error occurred, the location in the computer code of the application where the error occurred, an error code indicating a type of error, and the severity of the error." See Rock col. 3, lines 54-59. To analyze a message, Rock "queries a table or binary decision tree with at least some of the information sent in the error message to determine what action to take." See Rock col. 3, lines 64-67. Rock does not generate human readable statements based upon the error message.

Kleinman discloses a method for interpreting exceptions in a distributed object system. This is not equivalent to analyzing chassis codes from system firmware and software. Kleinman's method for translating an exception identifier into a text string uses the exception identifier to look up the text string in a table. See Kleinman col. 14, lines 1– 43. As the exception identifier indicates an error, it is not equivalent to chassis codes of the immediate application. Moreover, Kleinman does not process sequences of event codes to identify a particular problem.

For the purpose of the following discussion, the Examiner is again respectfully reminded of the basic considerations which apply to obviousness rejections.When applying 35 U.S.C. §103, the following tenets of patent law must be adhered to:

(A) The claimed invention must be considered as a whole;

(B) The references must be considered as a whole and must suggest the desirability and thus the obviousness of making the combination;

(C) The references must be viewed without the benefit of impermissible hindsight vision afforded by the claimed invention; and

(D) Reasonable expectation of success is the standard with which obviousness is determined. MPEP §2141.01, Hodosh v. Block Drug Co., Inc., 786 F.2d 1136, 1134 n.5, 229 USPQ 182, 187 n.5 (Fed. Cir. 1986).

In addition, it is respectfully noted that to substantiate a *prima facie* case of obviousness, the initial burden rests with the Examiner who must fulfill three requirements. More specifically:

<u>**First**</u>, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the references or to combine reference teachings.

<u>**Second**</u>, there must be a reasonable expectation of success.

<u>**Finally**</u>, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The ***teaching or suggestion*** to make the claimed combination **and** the ***reasonable expectation of success*** **must** ***both be found in the prior art,*** **and not** ***based on applicant's disclosure***. (emphasis and formatting added) MPEP § 2143, *In re vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991)

Amended claim 1 recites a method for analyzing chassis logs from electronic architecture, including the steps of:

a) automatically processing text strings associated with the chassis logs for one entity of the electronic architecture; and
b) transforming sequences of the text strings into human interpretable statements summarizing at least one problem detail of the entity;
c) wherein the chassis logs are specific to boot-up and operation of the electronic architecture.

Rock does not disclose processing text strings associated with chassis logs for one entity of the electronic architecture, as required by step a) of claim 1. As noted above, the error code of Rock is not equivalent to the chassis code of the immediate application. Rock does not disclose text strings associated with chassis logs. Instead, Rock discloses "the error message can contain one or more of the following types of information: the application in the ultrasounds system in which the error occurred, the location in the computer code of the application in which the error occurred, an error code indicating the type of error, and the severity of the error." See Rock col. 3, lines 54-59. Rock further discloses processing a single error code, not a sequence. As disclosed in paragraphs [0020-0028] of the specification, chassis codes (which may take the form of two 64-bit numbers) may be extracted from an electronic architecture, filtered and converted into text strings (e.g., hex strings representing chassis codes) by getcc section 102. Clearly, the error message of Rock is not equivalent to a chassis code converted into one or more text strings as taught by the immediate application.

Kleinman also does not disclose processing text strings associated with chassis logs for one entity of the electronic architecture, as required by step a). As argued above, Kleinman performs a look up of an exception identifier to retrieve a text string. See Kleinman col. 14, lines 1– 43. The exception identifier of Kleinman is not equivalent to a chassis code converted into one or more text strings as taught by the immediate application.

Step b) of claim 1 requires transforming sequences of the text strings to human interpretable statements summarizing at least one problem detail of the entity. Neither Rock nor Kleinman disclose or suggest transforming sequences of text strings into human interpretable statements summarizing at least one problem detail of the one entity. As previously discussed, Kleinman discloses interpreting exceptions in a distributed object computing environment. Fundamental to operation of the Kleinman system is the use of an exception identifier that uniquely identifies an exception raised by a remotely located device within the distributed object computing environment. Kleinman discloses a "computing environment to convert an exception identifier ... to a more readable message string." See Kleinman col. 4, lines 15-20. Kleinman, in col. 3, lines 7-9, specifically identifies "two basic types of exceptions: system exceptions

and user exceptions." "System exceptions are raised when errors are detected in the infrastructure of the object management facility," and "user exceptions are defined as part of the interface to an object, and are used to report errors that might be expected to occur during servicing of specified requests to that object." See Kleinman, col. 3, lines 17-21. Kleinman thus specifically describes exceptions associated with a client/server software environment. An exception identifier, as disclosed by Kleinman, is clearly not the same as chassis logs generated by electronic architecture.

Rock discloses, "after receiving the error message, the network management station 45 automatically analyzed the error message." See Rock col. 3, lines 62-64. Clearly Rock also only processes one error message at a time. Further, as argued above, the error message of Rock can contain one or more of: "the application in the ultrasound system in which the error occurred, the location in the computer code of the application where the error occurred, an error code indicating a type of error, and the severity of the error." See Rock col. 3, lines 54-59. Clearly these are not the same as chassis logs generated by electronic architecture as required by step b) of claim 1.

As described in paragraphs [0033-0036] of the specification and FIG. 4A and FIG. 4B of the drawings, text strings are processed by an analyzer that transforms sequences of text strings into human interpretable statements summarizing at least one problem detail of the entity, as in step b) of claim 1. In particular, paragraph [0033] teaches that for a chassis code of an entity, in step 306 of FIG. 4A, problem detail files are loaded. A sequence of text strings associated with chassis codes for a particular entity of the electronic architecture are compared to these problem detail files to identify a particular problem of the entity. An analyzer (e.g., one or more of analyzers 120, FIG. 2) uses problem detail files to match a sequence of chassis codes and identify a specific problem, resulting in a detailed summary describing the problem. See at least paragraphs [0033-0034]. Neither Rock nor Kleinman disclose processing a sequence of text strings.

Therefore, even when combined, Rock and Kleinman cannot render claim 1 obvious. Reconsideration of claim 1 is respectfully requested.

Claims 2-4 and 6-14 depend from claim 1 and benefit from like argument; but, in addition, these claims have other features that patentably distinguish over Rock in view of Kleinman. For example, claim 2 recites transforming the text strings to an

English statement setting forth one or more of problems and system health of the architecture. Neither Rock nor Kleinman discloses or suggests transforming text strings into an English statement setting forth one or more of problems and system health of the architecture. As argued above, Rock processes error codes one at a time and Kleinman processes exception identifiers one at a time. Therefore, Rock and Kleinman only indicate problems based upon single error events. Further, since the error code of Rock and the exception identifier of Kleinman indicate errors, neither Rock nor Kleinman indicate system health.

Claim 3 recites processing the text strings according one or more entities associated with the text string, the entities selected from the group of firmware, software, processors, architecture monitors, power monitors, cabinet monitors and I/O drivers. Neither Rock nor Kleinman disclose or suggest processing text string according to entities associated with the text strings.

Claim 4 recites processing text strings representative of one or more chassis code of the one or more entities. As argued above, neither Rock nor Kleinman disclose chassis codes or text strings representative of chassis codes.

Claim 6 recites processing problem detail of the chassis codes. Neither Rock nor Kleinman disclose processing problem detail of the chassis codes.

Claim 7 recites executing an embedded program with one of the chassis codes as an argument, to further analyze problems associated with the one entity. Neither Rock not Kleinman disclose executing a embedded program with one of the chassis codes as an argument to further analyze problems associated with the entity.

Claim 10 recites acquiring the text strings from an extraction tool coupled to the architecture. Neither Rock nor Kleinman disclose an extraction tool coupled to the architecture.

Claim 11 recites the extraction tool extracting the chassis logs from the architecture, separating the chassis logs according to the entities, and transforming the chassis logs to one or more text strings. Neither Rock not Kleinman discloses extracting chassis logs from the architecture.

Claim 12 recites accessing one or more analyzers coupled to the extraction tool. Neither Rock nor Kleinman disclose accessing one or ore analyzers coupled to the extraction tool.

Claim 13 recites utilizing a graphical user interface coupled to one or more of the analyzers. Neither Rock not Kleinman discloses a graphical interface coupled to one or more analyzers.

Claim 14 recites each of the analyzers processes text strings associated with one of the entities. Neither Rock nor Kleinman discloses an analyzer processing text strings associated with a entity.

For at least these reasons, Rock in view of Kleinman cannot render claims 2-4 and 6-14 obvious. Reconsideration of claim 2-4 and 6-14 is respectfully requested.

Amended claim 15 recites a system for analyzing text strings associated with chassis logs from electronic architecture, the architecture of the type having one or more entities generating the chassis logs, including:

a) one or more analyzers for analyzing sequences of text strings and for producing a human interpretable statement about one or more of the chassis logs, each of the analyzers associated with one of the entities selected from the group of firmware, software, processors, architecture monitors, power monitors, cabinet monitors, and I/O drivers; and
b) an interface for coupling the analyzers to an extraction tool acquiring the chassis logs from the architecture.

As argued above, the combination of Rock and Kleinman do not disclose analyzers that analyze sequences of text strings as required by step a) of claim 15. Further, neither Rock nor Kleinman disclose an interface for coupling analyzers to an extraction tool. Therefore, Rock in view of Kleinman cannot render claim 15 obvious.

Reconsideration of claim 15 is respectfully requested.

Claims 16-20 depend from claim 15 and benefit from like argument. However, these claims have additional features that distinguish over Rock in view of Kleinman. For example, claim 16 recites that the chassis logs derive from one or more of the entities. As argued above, neither Rock nor Kleinman disclose processing chassis codes.

Claim 17 recites an extraction tool coupled to the interface, the extraction tool extracting the chassis logs from the architecture, separating the chassis logs according to the entities, and transforming the chassis logs to one or more of the text strings. Again, as argued above, neither Rock nor Kleinman disclose extracting chassis logs

from the architecture. For example, in Rock, the error code is sent from the medical diagnostic ultrasound imaging system to a processor. See Rock col. 1, lines 49-52. Kleinman discloses that an exception identifier is received by a host computing system from a remote device. See Kleinman Abstract.

Claim 18 recites that the text strings include problem detail and chassis code. As argued above, neither Rock nor Kleinman disclose chassis codes. Further, neither Rock nor Kleinman disclose text strings containing problem detail.

Claim 19 recites that the problem detail includes an embedded program executable to perform further analysis of the text strings. Neither Rock nor Kleinman disclose an embedded program for performing further analysis of text strings.

For at least these reasons, Rock in view of Kleinman cannot render claims 16-20 obvious. Reconsideration of claims 16-20 is respectfully requested.

Claims 5 stands rejected under 35 U. S. C. §103(a) as being unpatentable over Rock in view of Kleinman and further in view of U. S. Patent Number 6,725,446 to Hahn (hereinafter "Hahn"). Respectfully we disagree.

Claim 5 depends from claim 1 and recites parsing each text string and sequentially processing each of the chassis codes. The Examiner states that Rock, Kleinman and Hahn are analogous art because they are from the same field of endeavor of electronic exception handling. Respectfully we disagree. The Examiner appears to have created this rejection based on 'piecemeal' findings in Hahn for the term 'parsing'. This piecemeal reference lacks motivation or suggestion for combination with Rock and/or Kleinman, as well as relevance to the present invention. The Court has held that "...every element of a claimed invention may often be found in the prior art. However, identification in the prior art of each individual part claimed is insufficient to defeat patentability of the whole claimed invention" *In re Rouffet,* 149 F.3d 1350, 1357, 47 USPQ2d 1453, 1457 (Fed. Cir 1988). Hahn, in fact, discloses in col. 1 lines 8-12 that its invention is directed to "a method and system for integrating heterogeneous information services, and more particularly to a method and system for assembling heterogeneous information streams with asynchronous or digital rates of arrival into a single information stream."

As argued above, Rock and Kleinman do not render claim 1 obvious, and even when combined with Hahn, do not render claim 5 obvious.

Reconsideration of claim 5 is respectfully requested.

In view of the above amendments and arguments, we contend that claims 1-20 are allowable and request reconsideration.

No additional fees are deemed due in connection with this amendment. If any fee is due, please charge Deposit Account No. 08-2025.

Respectfully submitted,

By: ______________________________

William A. Rudy, Reg. No. 34,916
LATHROP & GAGE L.C.
2345 Grand Boulevard, Suite 2400
Kansas City, Missouri 64108
Telephone: (816) 460-5819
Facsimile: (816) 292-2001